**B.H. Gardi College of Engineering & Technology**

**Name :** abc abc abc

**Enrollment No :** 2350306940XX

**E-mail ID :** 23mca0XX@gardividyapith.ac.in

**Subject :** Python

**Semester :** II

**Branch :** MCA

**Part I: Core Python [A]**

**P-01: Write a Python Program to Convert Celsius to Fahrenheit and vice –a-versa.**

* **Code :**

def celsius\_to\_fahrenheit(celsius):

return (celsius \* 9/5) + 32

def fahrenheit\_to\_celsius(fahrenheit):

return (fahrenheit - 32) \* 5/9

print("Temperature Converter")

print("=====================")

print("1. Celsius to Fahrenheit")

print("2. Fahrenheit to Celsius")

choice = input("Enter your choice (1 or 2): ")

if choice == '1':

celsius = float(input("Enter temperature in Celsius: "))

fahrenheit = celsius\_to\_fahrenheit(celsius)

print("Temperature in Fahrenheit:", fahrenheit)

elif choice == '2':

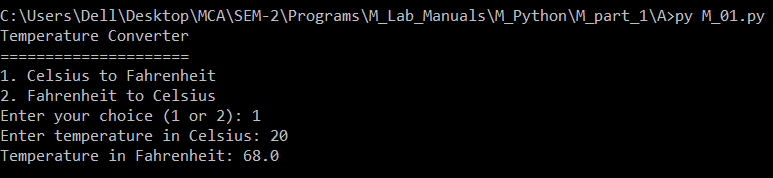
fahrenheit = float(input("Enter temperature in Fahrenheit: "))

celsius = fahrenheit\_to\_celsius(fahrenheit)

print("Temperature in Celsius:", celsius)

else:

print("Invalid choice")

* **Output :**

**P-02: Write a program in python to swap two variables without using temporary variable.**

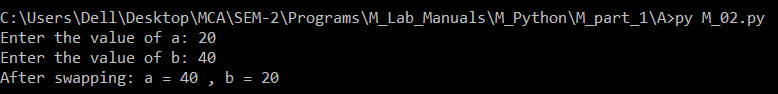
* **Code :**

a = input("Enter the value of a: ")

b = input("Enter the value of b: ")

a, b = b, a

print("After swapping: a =", a, ", b =", b)

* **Output :**

**P-03: Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal.**

* **Code :**

def decimal\_to\_binary(decimal):

return bin(decimal)[2:]

def decimal\_to\_octal(decimal):

return oct(decimal)[2:]

def decimal\_to\_hexadecimal(decimal):

return hex(decimal)[2:]

decimal\_number = int(input("Enter a decimal number: "))

binary = decimal\_to\_binary(decimal\_number)

octal = decimal\_to\_octal(decimal\_number)

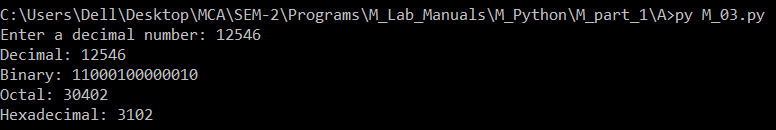
hexadecimal = decimal\_to\_hexadecimal(decimal\_number)

print("Decimal:", decimal\_number)

print("Binary:", binary)

print("Octal:", octal)

print("Hexadecimal:", hexadecimal)

* **Output :**

**P-04: Write a program to make a simple calculator (using functions).**

* **Code :**

def add(x, y):

return x + y

def subtract(x, y):

return x - y

def multiply(x, y):

return x \* y

def divide(x, y):

if y == 0:

return "Error! Division by zero is not allowed."

else:

return x / y

print("Simple Calculator")

print("=================")

print("Operations:")

print("1. Addition")

print("2. Subtraction")

print("3. Multiplication")

print("4. Division")

choice = input("Enter choice (1/2/3/4): ")

if choice in ('1', '2', '3', '4'):

num1 = float(input("Enter first number: "))

num2 = float(input("Enter second number: "))

if choice == '1':

print("Result:", add(num1, num2))

elif choice == '2':

print("Result:", subtract(num1, num2))

elif choice == '3':

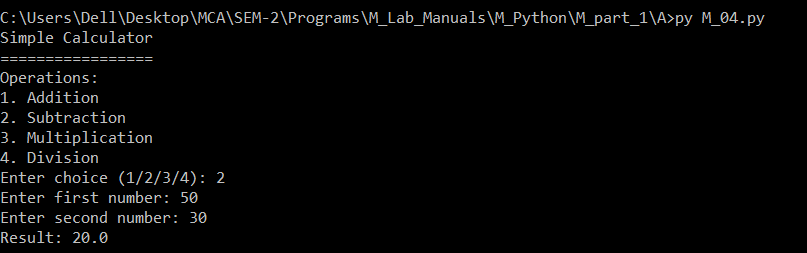
print("Result:", multiply(num1, num2))

elif choice == '4':

print("Result:", divide(num1, num2))

else:

print("Invalid Input")

****

* **Output :**

**P-05: Write a program in python to find out maximum and minimum number out of three user entered number.**

* **Code :**

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

num3 = float(input("Enter the third number: "))

maximum = num1

if num2 > maximum:

maximum = num2

if num3 > maximum:

maximum = num3

minimum = num1

if num2 < minimum:

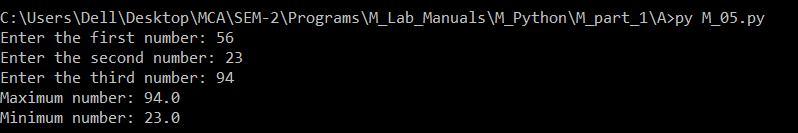
minimum = num2

if num3 < minimum:

minimum = num3

print("Maximum number:", maximum)

print("Minimum number:", minimum)

* **Output :**

**P-06: Write a program which will allow user to enter 10 numbers and display largest odd number from them. It will display appropriate message in case if no odd number is found.**

* **Code :**

numbers = []

for i in range(10):

number = float(input("Enter number {}: ".format(i + 1)))

numbers.append(number)

largest\_odd = None

for number in numbers:

if number % 2 != 0:

if largest\_odd is None or number > largest\_odd:

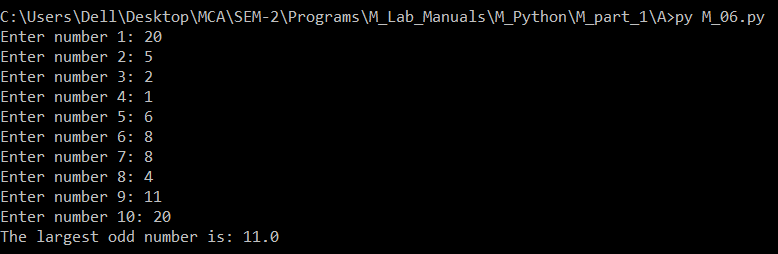
largest\_odd = number

if largest\_odd is not None:

print("The largest odd number is:", largest\_odd)

else:

print("No odd numbers found.")

* **Output :**

**P-07: Write a Python program to check if the number provided by the user is an Armstrong number.**

* **Code :**

def is\_armstrong(number):

num\_str = str(number)

num\_digits = len(num\_str)

armstrong\_sum = sum(int(digit) \*\* num\_digits for digit in num\_str)

return armstrong\_sum == number

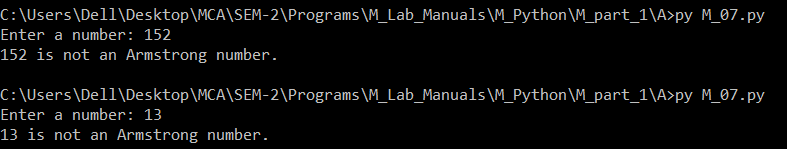
user\_number = int(input("Enter a number: "))

if is\_armstrong(user\_number):

print(user\_number, "is an Armstrong number.")

else:

print(user\_number, "is not an Armstrong number.")

* **Output :**

**P-08: Write a Python program to check if the number provided by the user is a palindrome or not.**

* **Code :**

def is\_palindrome(number):

num\_str = str(number)

return num\_str == num\_str[::-1]

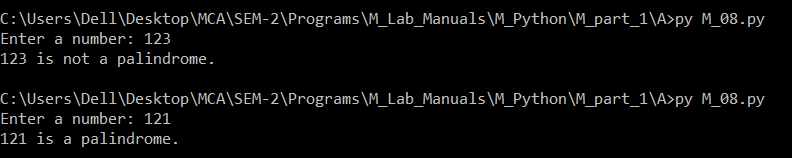
user\_number = input("Enter a number: ")

if is\_palindrome(user\_number):

print(user\_number, "is a palindrome.")

else:

print(user\_number, "is not a palindrome.")

* **Output :**

**P-09: Write a Python program to perform following operation on given string input: a) Count Number of Vowel in given string b) Count Length of string (do not use Len ()) c) Reverse string d) Find and replace operation e) check whether string entered is a palindrome or not.**

* **Code :**

user\_string = input("Enter a string: ")

vowel\_count = sum(1 for char in user\_string if char.lower() in "aeiou")

print("Number of vowels:", vowel\_count)

length = sum(1 for \_ in user\_string)

print("Length of string:", length)

reversed\_str = user\_string[::-1]

print("Reversed string:", reversed\_str)

find\_str = input("Enter the string to find: ")

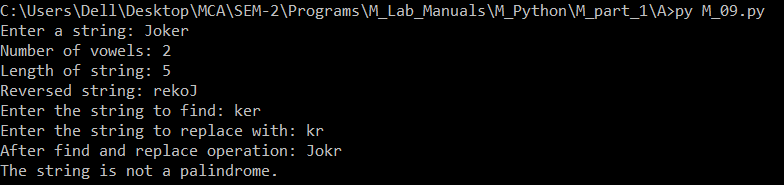
replace\_str = input("Enter the string to replace with: ")

replaced\_str = user\_string.replace(find\_str, replace\_str)

print("After find and replace operation:", replaced\_str)

is\_palindrome = user\_string == user\_string[::-1]

print("The string is a palindrome." if is\_palindrome else "The string is not a palindrome.")

* **Output :**

**P-10: Define a procedure histogram () that takes a list of integers and prints a histogram to the screen. For example, histogram ([4, 9, 7]) should print the following:**

**\*\*\*\***

**\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\***

* **Code :**

import matplotlib.pyplot as plt

def histogram(numbers):

plt.barh(range(len(numbers)), numbers)

plt.xlabel('Value')

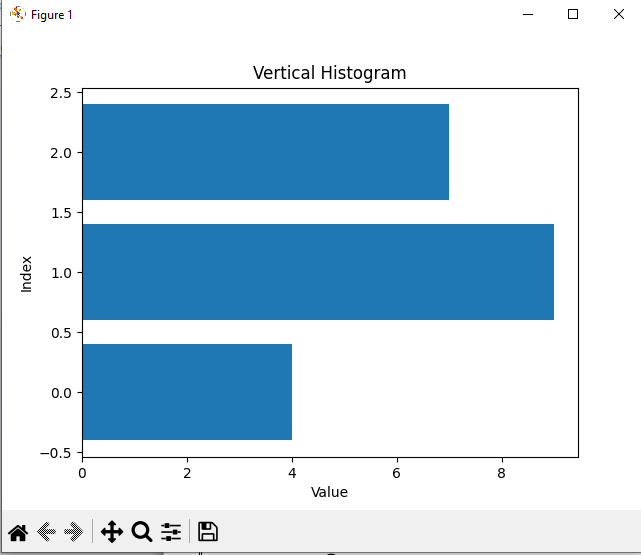
plt.ylabel('Index')

plt.title('Vertical Histogram')

plt.show()

histogram([4, 9, 7])

* **Output :**

****

**P-11: Write a program in python to implement Fibonacci series up to user entered number. (Use recursive Function).**

* **Code :**

def fibonacci(n):

if n <= 1:

return n

else:

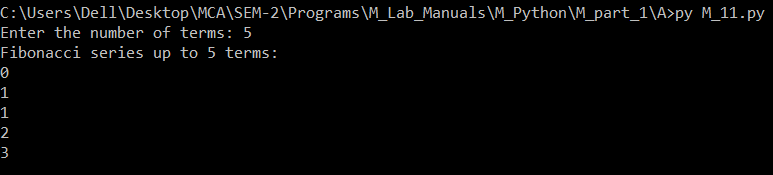
return fibonacci(n-1) + fibonacci(n-2)

num\_terms = int(input("Enter the number of terms: "))

print("Fibonacci series up to", num\_terms, "terms:")

for i in range(num\_terms):

print(fibonacci(i))

* **Output :**

**P-12: Write a program in python to implement Factorial series up to user entered number. (Use recursive Function).**

* **Code :**

def factorial(n):

if n == 0:

return 1

else:

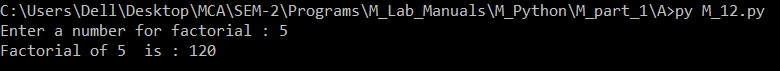
return n \* factorial(n - 1)

# Taking user input

num = int(input("Enter a number for factorial : "))

# Calculating and printing factorial

print("Factorial of", num, " is :", factorial(num))

* **Output :**

**P-13: Write a program in Python to implement readline, readlines, write line and writelines file handling mechanisms.**

* **Code :**

# Writing to a file

with open("demo.txt", "w") as file:

file.write("Hello\n")

file.write("World\n")

file.write("Python\n")

file.write("Programming\n")

# Reading from a file using readline()

print("Reading from file using readline():")

with open("demo.txt", "r") as file:

line = file.readline()

while line:

print(line.strip()) # strip() to remove newline character

line = file.readline()

print()

# Reading from a file using readlines()

print("Reading from file using readlines():")

with open("demo.txt", "r") as file:

lines = file.readlines()

for line in lines:

print(line.strip()) # strip() to remove newline character

print()

# Writing lines to a file using write()

with open("demo.txt", "a") as file:

file.write("This is a new line written using write()\n")

# Reading from a file after appending

print("Reading from file after appending:")

with open("demo.txt", "r") as file:

lines = file.readlines()

for line in lines:

print(line.strip()) # strip() to remove newline character

print()

# Writing lines to a file using writelines()

with open("demo.txt", "a") as file:

lines\_to\_write = ["Another line\n", "Yet another line\n", "One more line\n"]

file.writelines(lines\_to\_write)

# Reading from a file after writelines()

print("Reading from file after writelines():")

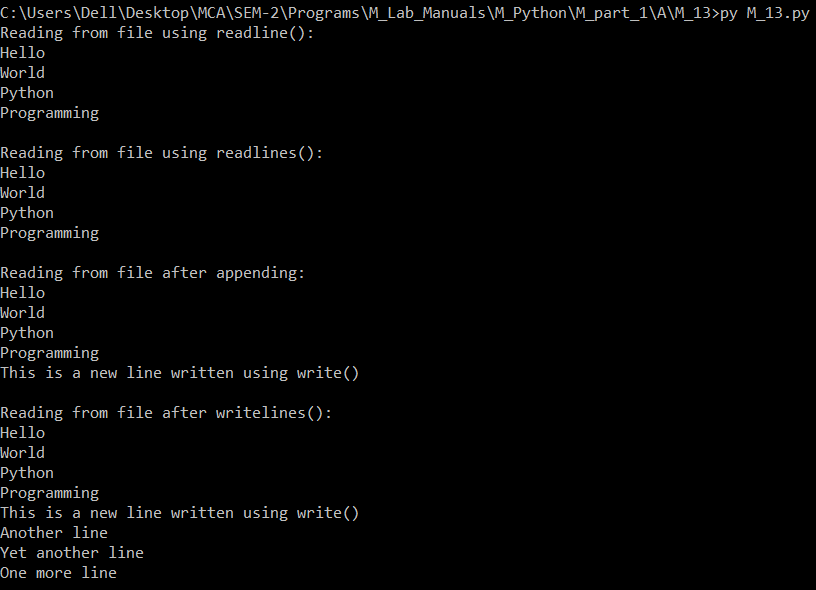
with open("demo.txt", "r") as file:

lines = file.readlines()

for line in lines:

print(line.strip()) # strip() to remove newline character

* **Output :**

****

**Part I: Core Python [B]**

**P-14: Write a program in python to implement Salary printing file read operation. (File format: Employee No, name, deptno, basic, DA, HRA, Conveyance) should perform below operations. a) Print Salary Slip for given Employee Number b) Print Employee List for Given Department Number.**

* **Code :**
  + **File name : empData.csv**

EmployeeNo,amploy,DeptNo,Basic,DA,HRA,Conveyance

1001,Manthan,100,50000,15000,10000,5000

1002,Nitesh,101,45000,12000,8000,4000

1003,Shivraj,102,48000,13000,9000,4500

1004,Prisha,100,52000,16000,11000,5500

1005,Punita,101,47000,12500,8500,4250

* + **File name : M\_14.py**

import csv

def read\_employee\_data(file\_name):

employee\_data = {}

with open(file\_name, newline='') as file:

reader = csv.DictReader(file)

for row in reader:

emp\_no = row['EmployeeNo']

employee\_data[emp\_no] = row

return employee\_data

def print\_salary\_slip(emp\_no, employee\_data):

if emp\_no in employee\_data:

emp\_details = employee\_data[emp\_no]

print("Salary Slip for Employee Number:", emp\_no)

print("Name:", emp\_details['amploy'])

print("Department Number:", emp\_details['DeptNo'])

print("Basic Salary:", emp\_details['Basic'])

print("Dearness Allowance (DA):", emp\_details['DA'])

print("House Rent Allowance (HRA):", emp\_details['HRA'])

print("Conveyance:", emp\_details['Conveyance'])

total\_salary = sum(float(emp\_details[field]) for field in ['Basic', 'DA', 'HRA', 'Conveyance'])

print("Total Salary:", total\_salary)

else:

print("Employee with Employee Number", emp\_no, "not found.")

def print\_employee\_list(dept\_no, employee\_data):

print("Employee List for Department Number:", dept\_no)

for emp\_no, details in employee\_data.items():

if details['DeptNo'] == dept\_no:

print("Employee Number:", emp\_no)

print("Name:", details['amploy'])

print("Basic Salary:", details['Basic'])

print()

# Read employee data from file

employee\_data = read\_employee\_data("empData.csv")

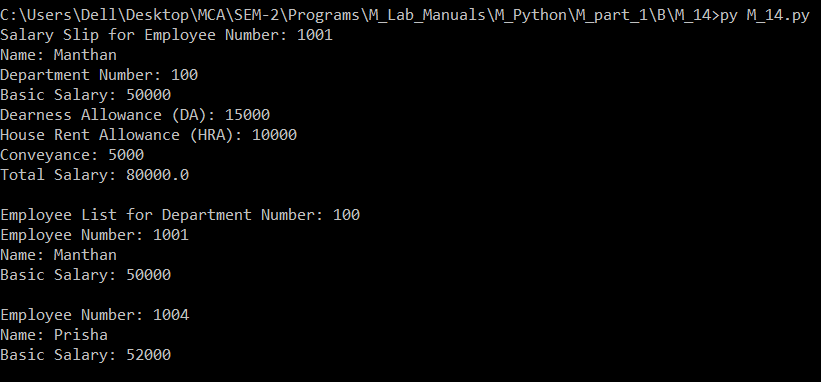
# Test cases

print\_salary\_slip("1001", employee\_data)

print()

print\_employee\_list("100", employee\_data)

* **Output :**

****

**P-16: Write a Python program to implement module.**

* **Code :**
  + **File name : My\_Module.py**

def greet(name):

return "Hello, " + name + "!"

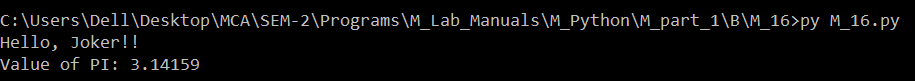
PI = 3.14159

* + **File name : M\_14.py**

import My\_Module as MM

print(MM.greet("Joker!"))

print("Value of PI:", MM.PI)

* **Output :**

**P-18: Write a program to read CSV file and generate output using HTML table.**

* **Code :**
  + **File name : empData.csv**

EmployeeNo,amploy,DeptNo,Basic,DA,HRA,Conveyance

1001,Manthan,100,50000,15000,10000,5000

1002,Nitesh,101,45000,12000,8000,4000

1003,Shivraj,102,48000,13000,9000,4500

1004,Prisha,100,52000,16000,11000,5500

1005,Punita,101,47000,12500,8500,4250

* + **File name : M\_18.py**

import csv

def generate\_html\_table(file\_name):

html\_output = "<html>\n<head>\n<title>CSV to HTML Table</title>\n</head>\n<body>\n"

html\_output += "<table border='1'>\n"

with open(file\_name, newline='') as file:

reader = csv.reader(file)

for row in reader:

html\_output += "<tr>\n"

for cell in row:

html\_output += "<td>{}</td>\n".format(cell)

html\_output += "</tr>\n"

html\_output += "</table>\n</body>\n</html>"

return html\_output

# Read CSV file and generate HTML table

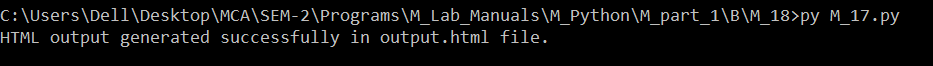
html\_content = generate\_html\_table("empData.csv")

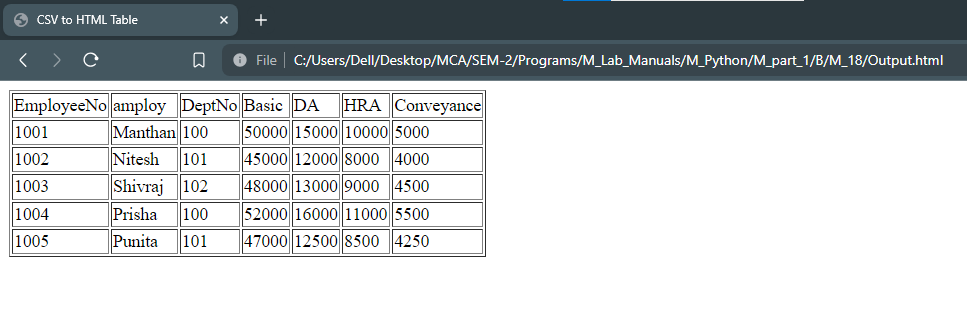
# Write HTML content to a file

with open("Output.html", "w") as html\_file:

html\_file.write(html\_content)

print("HTML output generated successfully in output.html file.")

* **Output :**

****

**P-19: Write a program to process CSV file using CSV module.**

* **Code :**
  + **File name : empData.csv**

EmployeeNo,amploy,DeptNo,Basic,DA,HRA,Conveyance

1001,Manthan,100,50000,15000,10000,5000

1002,Nitesh,101,45000,12000,8000,4000

1003,Shivraj,102,48000,13000,9000,4500

1004,Prisha,100,52000,16000,11000,5500

1005,Punita,101,47000,12500,8500,4250

* + **File name : M\_19.py**

import csv

def process\_csv(file\_name):

with open(file\_name, newline='') as file:

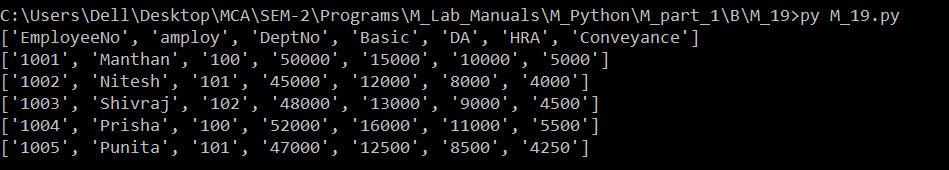
reader = csv.reader(file)

for row in reader:

print(row)

# Process the CSV file

process\_csv("empData.csv")

* **Output :**